Tool Support for the Goal-Oriented Language

Yi Chu
SITE, University of Ottawa
yi.chu@tycohealthcare.com

Overview

- GRL and GRL Editor
- Generic Modeling Environment (GME)
- GRL Editor Based on GME
- Meta-model Evolution Experiments
- Future Works

Goal-Oriented Requirement Language

- A language for supporting Goal-Oriented modeling and reasoning of non-functional requirements.
- Part of User Requirement Notation (URN, will be standardized as Z.150 by the ITU-T) that is used in early stages of software development.
- Complementary to UCM (Use Case Map for functional requirements analysis with scenarios).

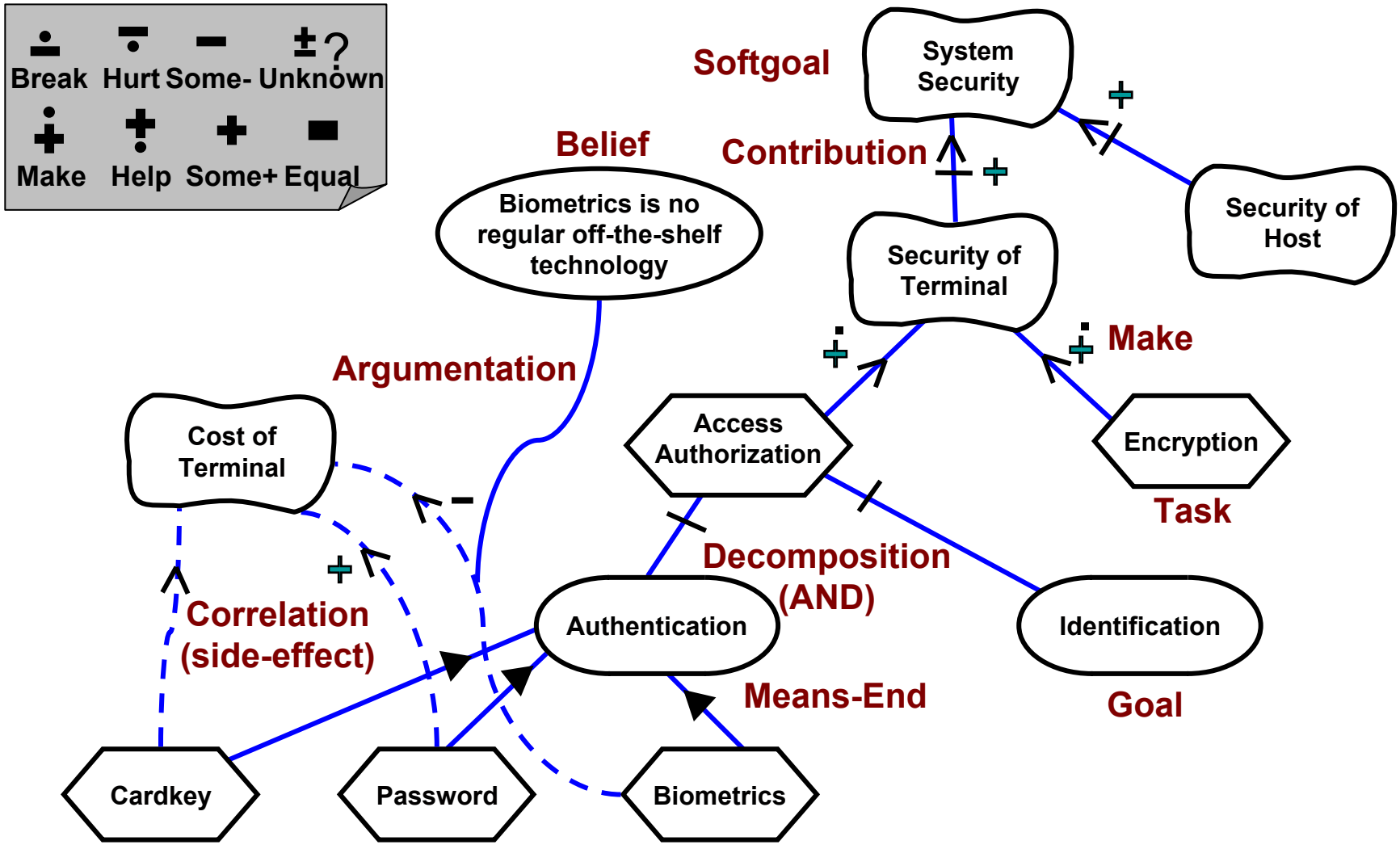
GRL Main Concepts

Three main types of concepts

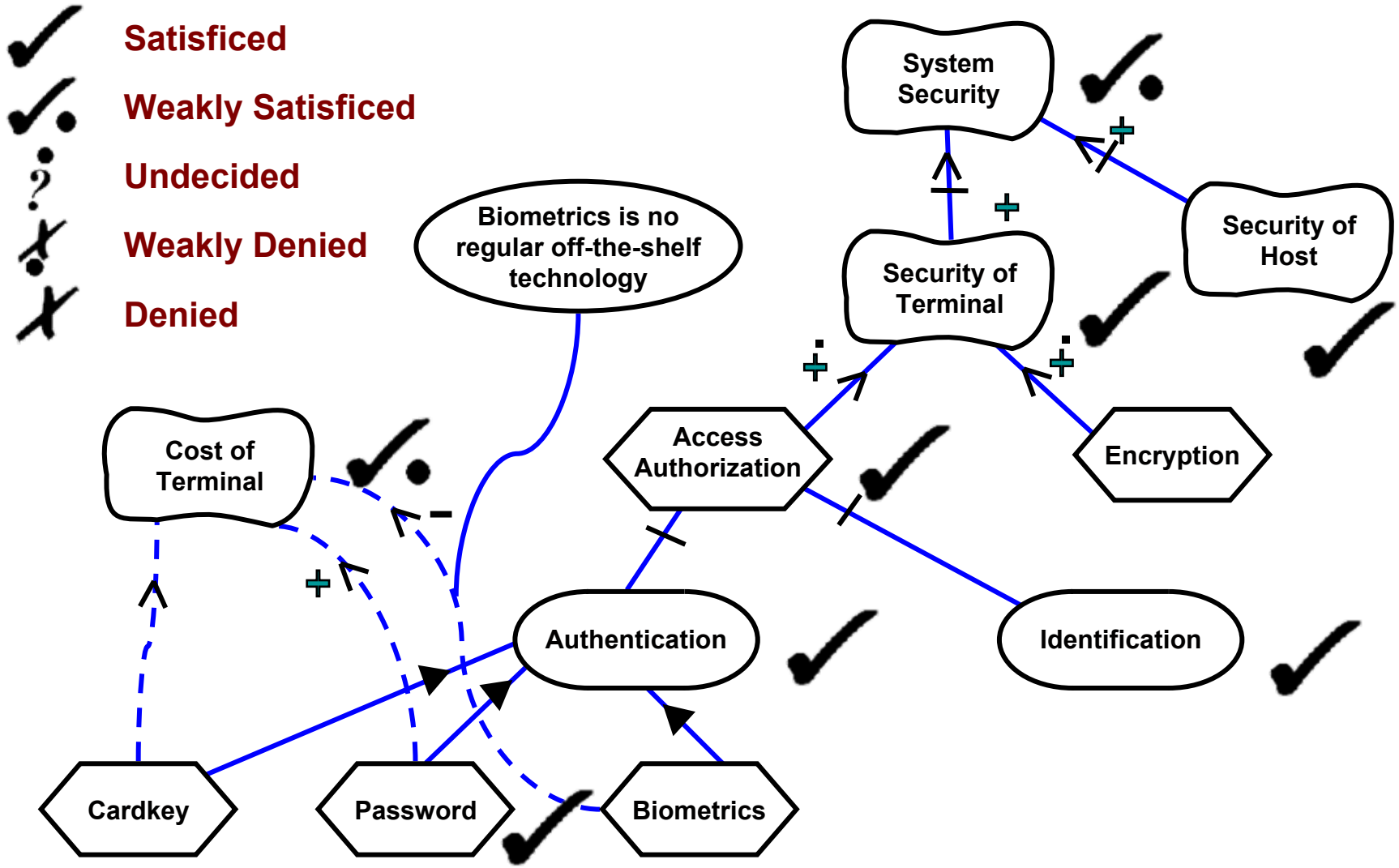
- Intentional Elements: Goal, SoftGoal, Task, and Belief.
- Links: Contribution, Correlation, Means-End, and Decomposition.
- Actors and Actor Boundary

Other concepts: Resource, And/Or links

Basic GRL Notations



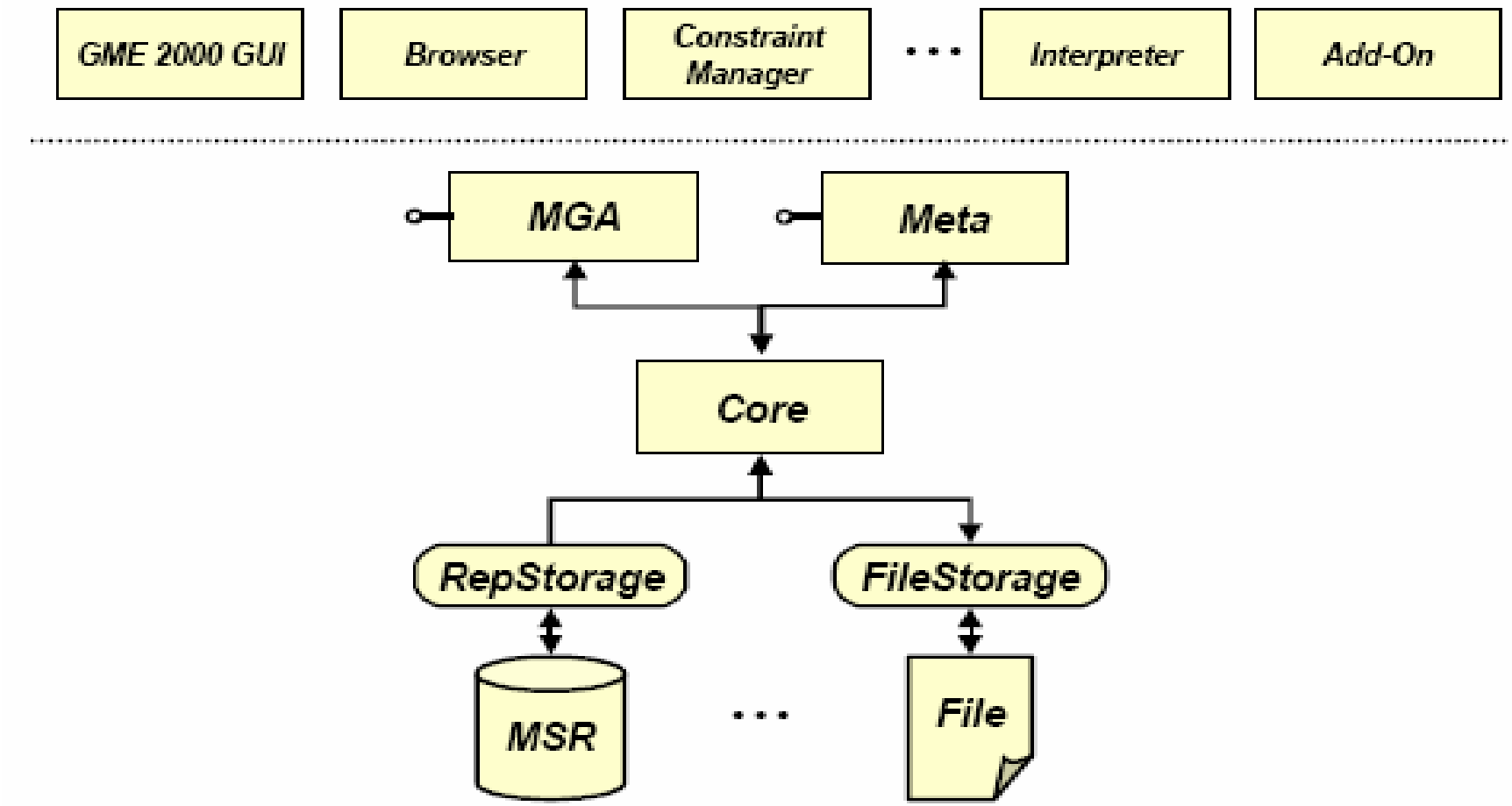
GRL Evaluation



Generic Modeling Environment (GME)

- Configurable toolset for creating domain-specific modeling environment
- Modeling language (Paradigm) is supported by meta-models
- Meta-models specify concepts for constructing the models, the relationships between the concepts and how they are organized
- Specify syntactic definition with UML class diagram (with extensions in information presentation)
- Specify static semantics with OCL

GME Architecture (I)



GME Architecture (II)

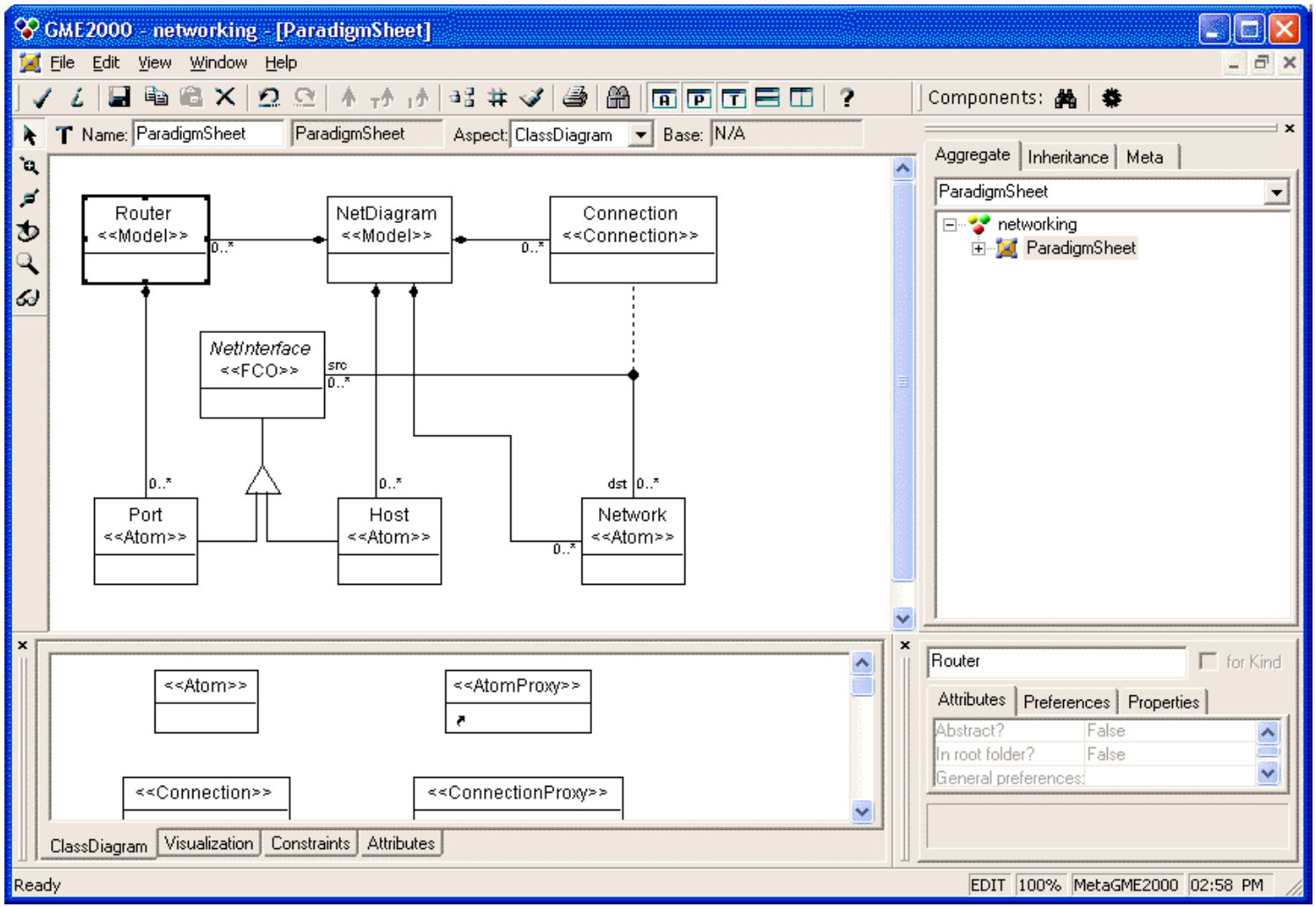
- Meta defines the modeling paradigm. MGA implements the GME modeling concepts for the given paradigm
- The user interacts with the components at the top layer
- Model interpreters perform translation and analysis of models (e.g. extract and translate semantic knowledge from the models)
- All components are COM servers
- Two high-level component interfaces

  Wrapper objects that mirror internal model objects and shield the user from the lower level details of the COM interface and provide support for easy traversal of the models

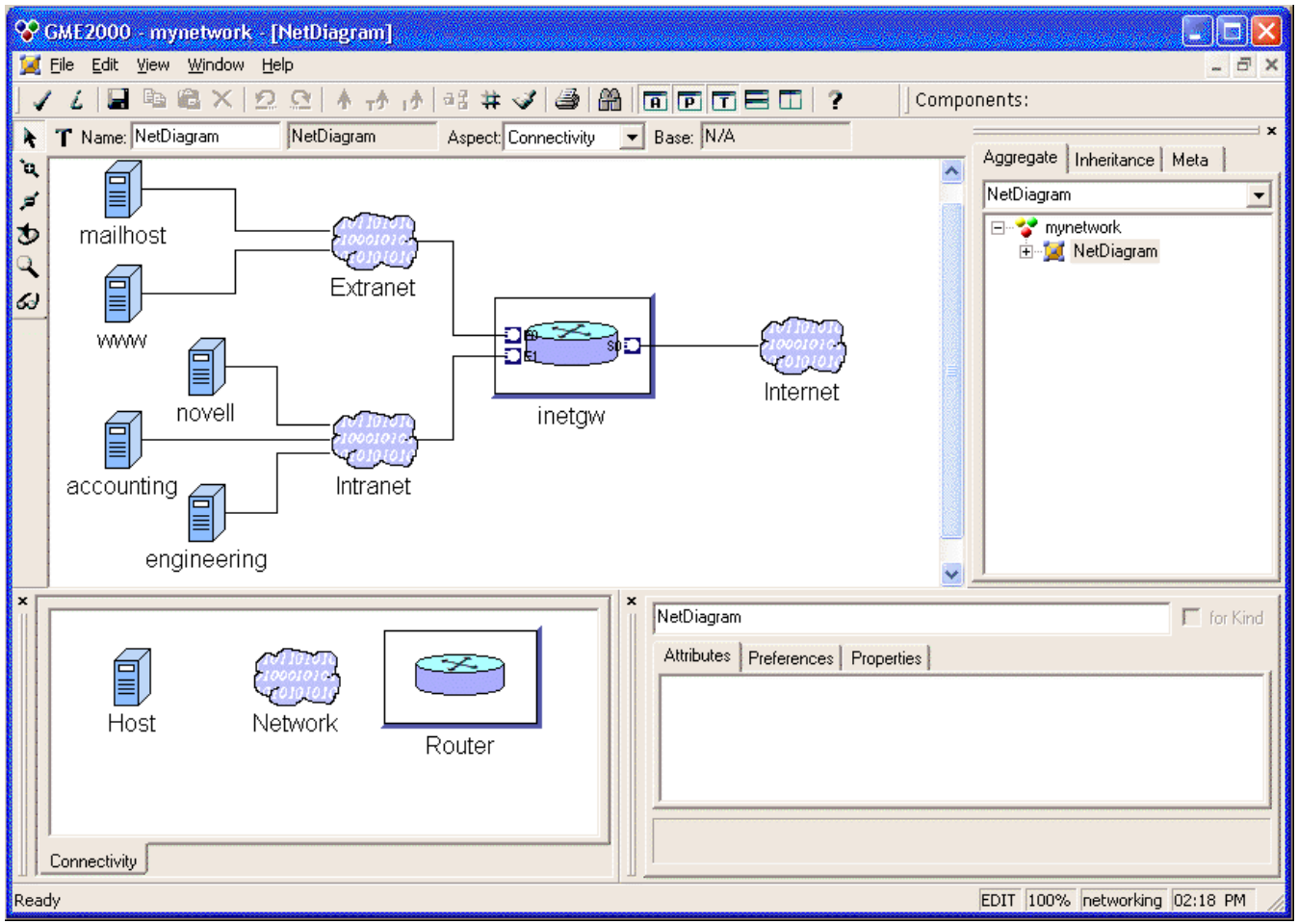
Basic Modeling Concepts

- Atom – elementary object.
- Model – Compound object that can have inner parts and structures
- Connection – Relationship between two objects (within one model)
- Reference – Refers to a real object in any place of the model hierarchy
- Set – Similar to an Aggregation in UML

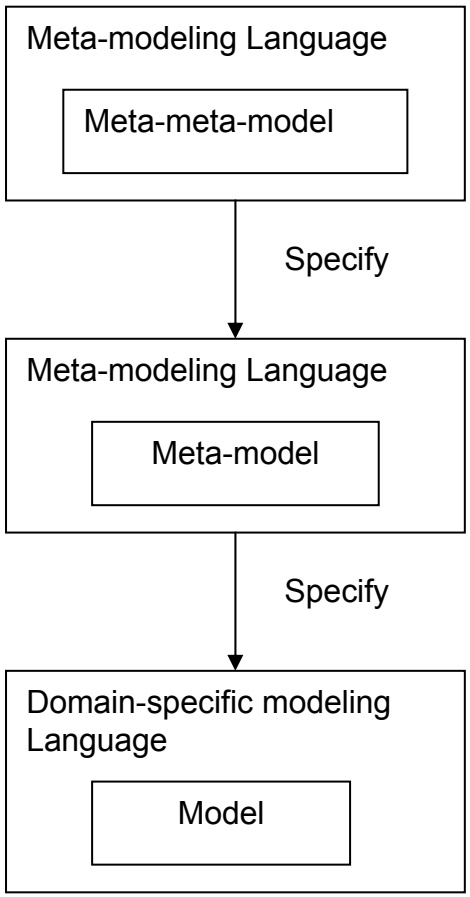
Construct a Meta-model in GME



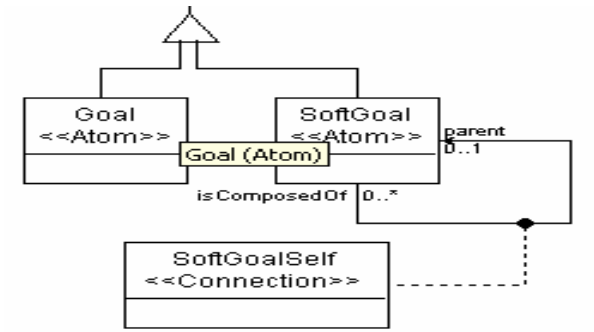
Create a Model in GME (I)



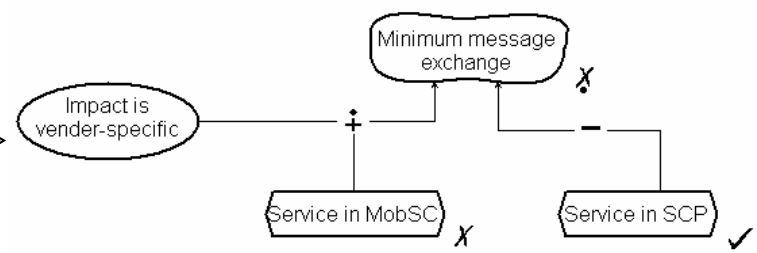
Create models in GME (II)



Example: GRL Editor Meta-model

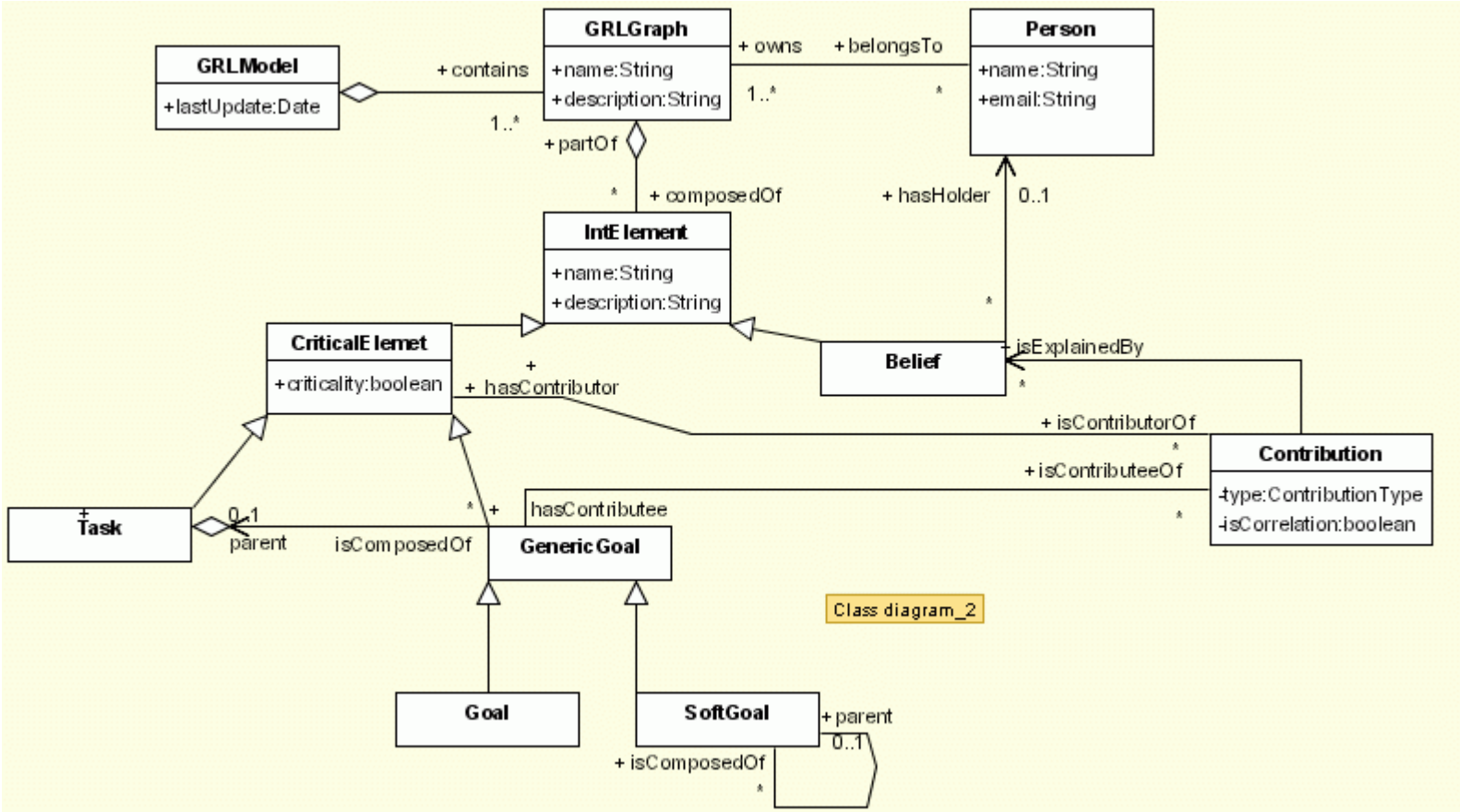


Example: GRL model

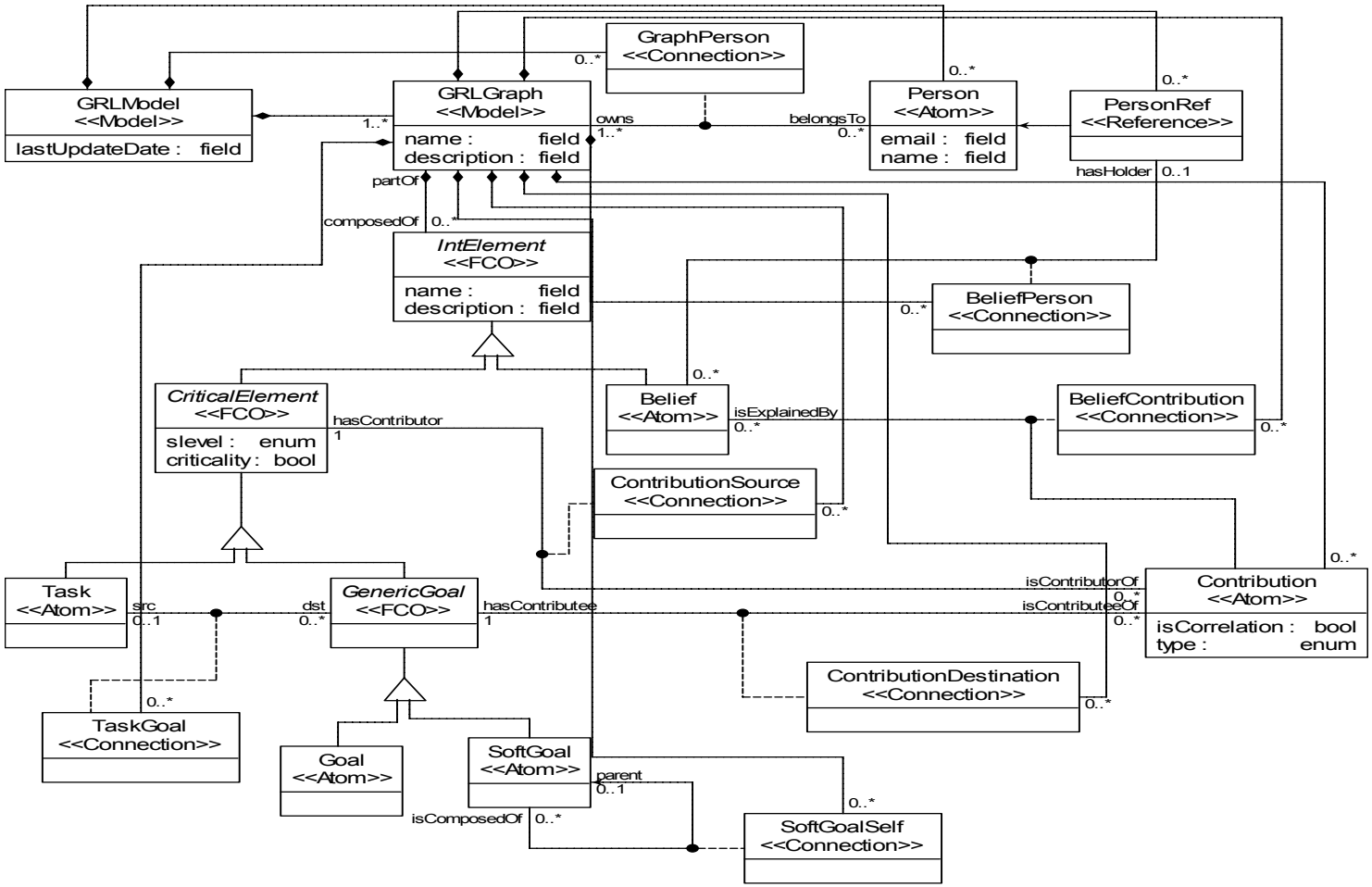


Abstract Definition of GRL Meta-model

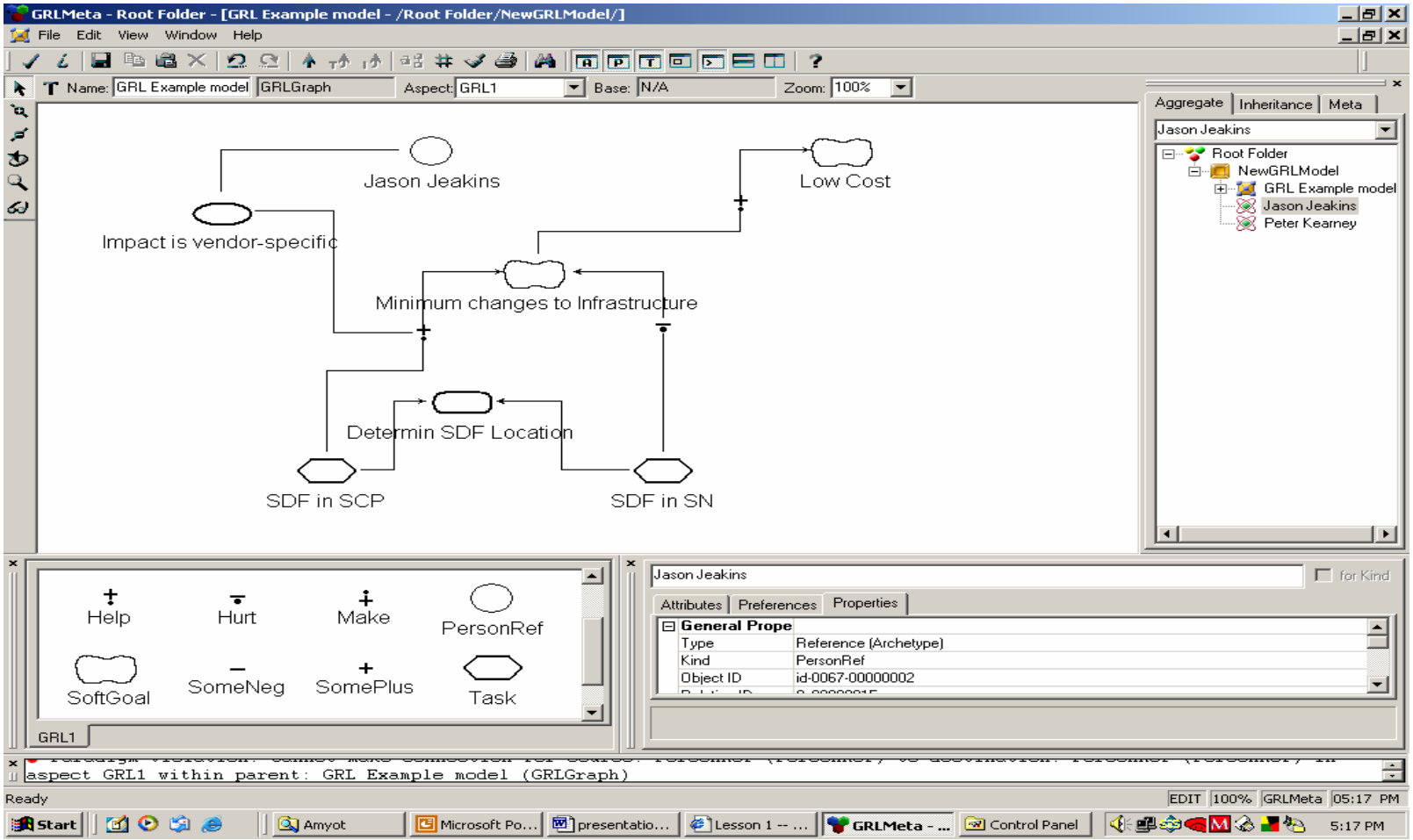
(Simplified Version using UML)



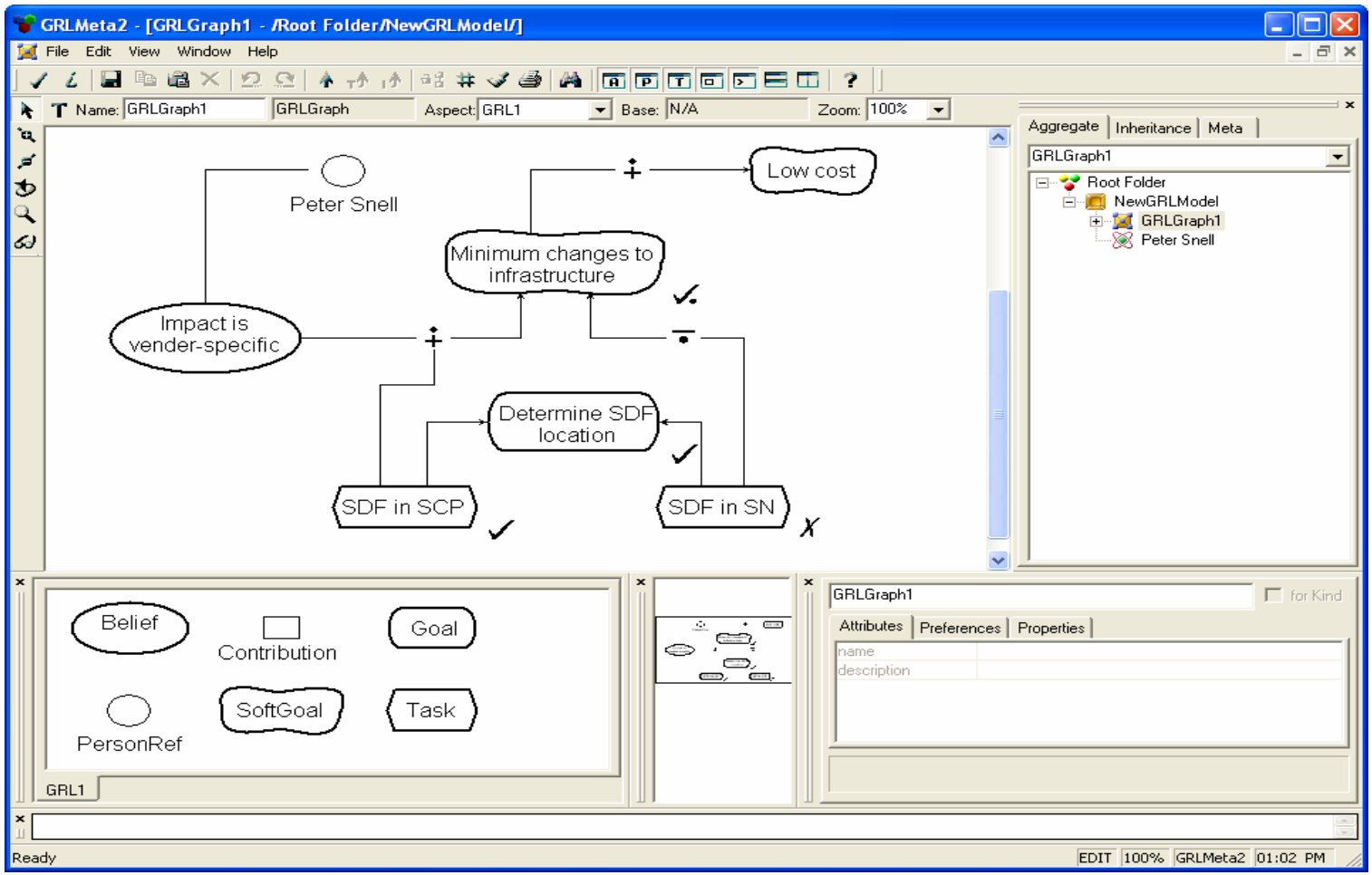
GRL Meta-model



GRL Editor (I)


GRLMeta - Root Folder - [GRL Example model - /Root Folder/NewGRLModel/]
File Edit View Window Help
Name: GRL Example model
GRLGraph
Aspect: GRL1
Base: N/A
Zoom: 100%
Jason Jeakins
Low Cost
Impact is vendor-specific
Minimum changes to Infrastructure
Determin SDF Location
SDF in SCP
SDF in SN
Aggregate
Inheritance
Meta
Root Folder
NewGRLModel
GRL Example model
Jason Jeakins
Peter Kearney
Help
Hurt
Make
PersonRef
SoftGoal
SomeNeg
SomePlus
Task
GRL1
Attributes
Preferences
Properties
General Prope
Type
Reference (Archetype)
Kind
PersonRef
Object ID
id-0067-00000002
aspect GRL1 within parent: GRL Example model (GRLGraph)
Ready
EDIT
100%
GRLMeta
05:17 PM
Start
Amyot
Microsoft Po...
presentatio...
Lesson 1 -- ...
GRLMeta - ...
Control Panel
5:17 PM

GRL Editor (II)
(with enhancement to Decorators)



Meta-model Evolution Experiments

- To study the behavior of the GRL model editor when the meta-model has changed
- Two meta-models used in each experiments, with one GRL model that is created from the old meta-model
- The experiment is marked as “Passed” if the GRL model can be opened with the new meta-model. Otherwise the result is “Failed”
- Targeted types - entity, link, attribute and reference

Experiment on Attribute Changes

No.	Description	Result	Explanations
1	Add a new Attribute of	Passed	The model is opened correctly with the new paradigm. New attribute of an object is set with its default value
2	Rename a used Attribute (the attribute is used in the model)	Failed	The model cannot open with the new paradigm
3	Rename an unused Attribute (the attribute is not used in the model)	Passed	
4	Delete a used Attribute	Passed	The model opens with the removed attribute not seen
5	Delete an unused Attribute	Passed	

Meta-model Evolution Experiments
(Conclusion)

An upgraded paradigm is backward compatible if

- No renaming or deleting of an element, link, reference and reference source that has been used in an existing model
- Adding, renaming, or deleting any type of objects that are not used in the meta-model

With the exception of the Attribute

- Deleting a used Attribute does not break the backward compatibility of the paradigm.

Future Works

- Add implementations for concepts such as Actor
- Add OCL constraints to the entities when more complex logical conditions need to apply to the meta-model
- Development of an GME Interpreter that performs GRL model evaluation

References

- Amyot, D.: Introduction to the User Requirements Notation: Learning by Example. In: *Computer Networks*, 42(3), 285-301, 21 June 2003.
- Bakay, A.: *The MGA library*. ISIS, Vanderbilt University http://www.cs.virginia.edu/~pnn7f/vest/docs/mgalib.pdf. Accessed May 2005.
- Chung, L., Nixon, B.A., Yu, E., & Mylopoulos, J.: *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, Dordrecht, USA, 2000.
- Eclipse: Graphical Modeling Framework (GMF), http://www.eclipse.org/gmf/
- IBM: *Rational Software Architecture*, http://www-306.ibm.com/software/awdtools/architect/swarchitect/
- Institute for Software Integrated Systems: *The Generic Modeling Environment (GME)*, 2004. http://www.isis.vanderbilt.edu/Projects/gme/
- ITU-T – International Telecommunications Union: Recommendation Z.150 (02/03), User Requirements Notation (URN) – Language Requirements and Framework. Geneva, Switzerland, 2003.
- ITU-T, URN Focus Group: *Draft Rec. Z.151 – GRL: Goal-oriented Requirement Language (GRL)*. Geneva, Switzerland, Sept. 2003. http://www.UseCaseMaps.org/urn/
- Janmohamed, J: *Expressing Goal-oriented Requirement Language in UML 2.0*. CSI 4900 project report, University of Ottawa, April 2005.
- jUCMNav: http://jkealey.shade.ca:82/twiki/bin/view/ProjetSEG/WebHome
- Knowledge Management Lab, University of Toronto: *Organization Modeling Environment*, http://www.cs.toronto.edu/km/ome/
- Kealey, J., Tremblay, E., Daigle, J.-P., McManus, J., Clift-Noël, O., and Amyot, D.: jUCMNav: une nouvelle plateforme ouverte pour l'édition et l'analyse de modèles UCM. To appear in: *Nouvelles TEchnnologies de la RÉpartition (NOTERE'05)*, Gatineau, Canada, August 2005.

References (II)

- Lamsweerde, A.v.: Requirements Engineering in the Year 00: A Research Per-spective. In: *Proc. of 22nd Intl Conference on Software Engineering (ICSE)*. Limerick, Ireland, ACM press, 2000.
- Liu, L., & Yu, E.: Designing Information Systems in Social Context: A Goal and Scenario Modelling Approach. *Information Systems (Journal)*, Vol.29, No.2. 2003. http://www.cs.toronto.edu/~liu/publications/
- OMG – Object Management Group (2003). *Unified Modeling Language Specification (UML)*, version 1.5, March 2003. http://www.omg.org/uml/
- Telelogic: *Telelogic TAU*, http://www.telelogic.com/products/tau/
- University of Toronto: *GRL Ontology,* http://www.cs.toronto.edu/km/GRL/
- University of Toronto: *Telos: Representing Knowledge about Information Systems,* http://www.cs.toronto.edu/~jm/2507S/Notes04/Telos.pdf
- URN Focus Group: *Draft Rec. Z.151 – Goal-oriented Requirement Language (GRL)*. Geneva, Switzerland, Sept. 2003. http://www.UseCaseMaps.org/urn/. See also http://www.cs.toronto.edu/km/GRL/
- Weiss, M., & Amyot, D.: Designing and Evolving Business Models with URN. *Montreal Conference on eTechnologies (MCeTech)*, Montréal, Canada, January 2005.
- Xactium Limited: XMF-Mosaic Getting Started Guide, Version 1.0, July 2005
- Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *3rd IEEE Int. Symp. on Requirements Engineering (RE'97)*, Washington, USA, 226-235, 1997.
- Yu, E., & Liu, L.: *Organization Modelling Environment* (OME), 2000. http://www.cs.toronto.edu/km/ome/
- Yu, E., & Mylopoulos, J.: Why goal-oriented requirements engineering. *Proceedings of the 4th REFSQ*, Pisa, Italy, 15–22, 1998.